UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Estruturas de Dados e Algoritmos

Ano Letivo de 2019/20

1º Teste de Frequência – VERSÃO D **13 de maio de 2020**

Duração: 50 min. Com consulta.

Problema 1 *(100%)*

Uma companhia aérea tem uma frota de aviões e faz vários voos em cada altura. O tipo de dados `struct voos` contém dados sobre os voos que decorrem num determinado instante de tempo:

- `nome`, string C com 25 carateres úteis, que contém o nome que identifica o avião que faz o voo, e.g. `"TAP-Gago Coutinho"`;
- `ano_construcao`, inteiro sem sinal que identifica o ano de construção do avião;
- `timestamp`, real com o número de segundos decorridos desde o início do voo;
- `destino`, *array* do tipo `unsigned char`, de dimensão 65, contendo o destino do voo.

a) Declare a estrutura `struct voos` e inicie a programação da função `main()` com uma tabela de 3 fichas contendo nos seus 3 primeiros campos valores predefinidos escolhidos por si, mas diferentes para cada ficha. O campo `destino` pode ser inicializado com o valor `"\0"` para todas as fichas. *(25%)*

b) Use a biblioteca `iomanip` para escrever a função `displayVoos()` que recebe como parâmetros uma tabela do tipo `struct voos` e a sua dimensão, e apresenta na consola os dados contidos nos seguintes campos dos elementos da tabela, formatados conforme indicado: `nome`, na coluna 1, alinhado à esquerda, 30 carateres; `ano_construcao`, alinhado à direita na coluna 45; `timestamp`, alinhado à direita na coluna 55, apresentado em notação fixa (use o formatador `fixed`) e com 1 casa decimal. *(35%)*

c) Escreva a função `getVoo()` que lê de um ficheiro em modo binário uma (e apenas uma) estrutura de dados do tipo `struct voos` situada no índice `i` do ficheiro (um inteiro `>=0`). A função recebe o nome do ficheiro (string C), o índice `i` e o parâmetro `voo` do tipo `struct voos` passado por referência, através do qual a função devolve os dados lidos no ficheiro. O ficheiro pode conter um número arbitrário de estruturas do tipo `struct voos` (incluindo zero estruturas). A função devolve (faz `return` de) `true` ou `false` consoante a operação de leitura é bem sucedida ou não, respetivamente. *(40%)*

Ao longo da realização do problema, escreva gradualmente uma função `main()` que deverá ir usando para testar a sua resposta às 3 alíneas. Para testar a função da alínea c), escreva na função main() o código necessário para guardar num ficheiro em modo binário os dados da tabela inicializada na alínea a) e apresentada na consola na alínea b). Este ficheiro servirá para chamar a função `getVoo()`.

UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Estruturas de Dados e Algoritmos

Ano Letivo de 2019/20

1º Teste de Frequência – VERSÃO D **13 de maio de 2020**

Duração: 50 min. Com consulta.

Problema 2 *(100% = 15% + 15% + 20% + 20% + 15% + 15%)*

A classe de objetos `CClock` permite definir uma hora em horas e minutos. Possui os atributos `protected` horas e minutos (todos do tipo int) e os métodos `public` seguintes: i) um construtor por defeito/omissão (objeto com 3:00); ii) um construtor por enumeração (recebe 2 inteiros, correspondentes às horas e minutos); iii) um construtor por cópia; iv) a sobrecarga do operador + (adianta a hora um determinado n.º de minutos passado como parâmetro). Nesta classe, os minutos devem ser sempre positivos e inferiores a 60. Nos métodos que realizam a sobrecarga dos operadores, são sempre efetuadas as correções necessárias para que aquelas condições se continuem a verificar.

a) Escreva a declaração da classe de objetos `CClock`. Nesta alínea, não deve definir qualquer método.
b) Defina os construtores por defeito e por enumeração.
c) Defina a sobrecarga do operador + (tenha em atenção a definição standard deste operador).
d) Escreva a declaração de uma classe derivada da classe anterior, `CViagem` que inclui mais um atributo `private` inteiro, `tempo`, representando a duração em minutos de uma viagem iniciada à hora definida na classe base. Este atributo pode ter um valor superior a `60`. Existem os métodos `public` da classe base e ainda o método `simultanea()` que verifica se uma viagem começa ou termina ao mesmo tempo que outra. Nesta alínea, não deve definir qualquer método.
e) Defina os construtores por defeito (objeto com hora 3:00 e duração 80 min) e por enumeração.
f) Defina o método `simultanea()`. Este deve chamar o operador + herdado.

Ao longo da realização do problema, escreva gradualmente uma função `main()` que deverá ir usando para testar a sua resposta às várias alíneas.

Teste o operador + com as horas 3:00 e 3:10, em que os valores dos minutos a adiantar são de 80 min e 70 min respetivamente.

Teste a função `simultanea()` com duas viagens iniciadas e com tempos de, respetivamente (3:00, 80 min) e (3:10, 70 min).